Computers and Mathematics with Applications 57 (2009) 1057–1072

Contents lists available at ScienceDirect

Computers and Mathematics with Applications

journal homepage: www.elsevier.com/locate/camwa

A cross-language focused crawling algorithm based on multiple relevance prediction strategies

Zhumin Chen [a,*], Jun Ma [a], Jingsheng Lei [b], Bo Yuan [c], Li Lian [a], Ling Song [a,d]

[a] School of Computer Science and Technology, Shandong University, Jinan 250061, China
[b] College of Information Science and Technology, Hainan University, Haikou 570228, China
[c] Department of Computer Science, University of Southern California Los Angeles, CA 90088, USA
[d] School of Computer Science and Technology, Shandong Jianzhu University, Jinan 250101, China

ARTICLE INFO

Keywords:
Focused crawling
Multiple relevance prediction strategies
Topic taxonomy
Cross-language
Shark-search algorithm

ABSTRACT

Focused crawling is increasingly seen as a solution to address the scalability limitations of existing general-purpose search engines, by traversing the Web to only gather pages that are relevant to a specific topic. How to predict the relevance of the unvisited pages pointed to by candidate URLs in the crawling frontier to a given topic is a key issue in the design of focused crawlers. In this paper, we propose a novel approach based on multiple relevance prediction strategies to address this problem. For cross-language crawling, we first introduce a hierarchical taxonomy to describe topics in both English and Chinese. We then present a formal description of the relevance predicting process and discuss four strategies that make use of page contents, anchor texts, URL addresses and link types of Web pages, respectively, to evaluate the relevance more accurately, in which we propose a particular strategy using Chinese URL addresses to estimate the relevance of cross-language Web pages. Finally, we get a new focused crawling algorithm (FCMRPS, Focused Crawling based on Multiple Relevance Prediction Strategies) based on the combination of these strategies and Shark-Search, which is a classic focused crawling algorithm. Experiments show that the FCMRPS is more effective than the traditional algorithms, namely Breadth-First, Best-First and Shark-Search, in terms of precision and sum of information.

1. Introduction

Due to the limited bandwidth, storage, computational resources and rapid growth of the World Wide Web, unprecedented scaling challenges have been posed for search engines. Although search engine technology has scaled dramatically to keep up with the growth of the Web, these general-purpose crawlers and search engines have presented some serious limitations as follows:

(1) It is impossible for them to index and analyze all pages and maintain comprehensive, up-to-date search indexes.
(2) They may return hundreds or more links to a user's query, however since they lack the understanding of the query the pages pointed to by these links may not closely relevant to the user's query.
(3) They cannot satisfy the query requests of different background, purpose and period.

* Corresponding author.
E-mail addresses: chenzhumin@mail.sdu.edu.cn (Z. Chen), majun@sdu.edu.cn (J. Ma), jshlei@hainu.edu.cn (J. Lei), boyuan@usc.edu (B. Yuan), lianli@sdu.edu.cn (L. Lian), song_ling@sdjzu.edu.cn (L. Song).

0898-1221/$ – see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.camwa.2008.09.021

(4) Dynamic contents, such as news and financial data, on the Web are growing and changed frequently. Many search engines may take up to one month for refreshing their indices on the full Web. Therefore, the query results may be not valid at the time that the query is issued.

Therefore, fast crawling technology is needed to gather the Web pages with high relevance and quality and keep them up to date. It is also necessary to add capabilities to search engines that respond to the particular information needs expressed by topics or interest profiles. So focused crawling is regard as a potential solution to overcome these limitations.

Focused crawlers traverse a subset of the Web to only gather pages that are relevant to a specific topic. An important assumption implicit in focused crawling is that the pages with respect to related topics tend to be neighbors of each other, i.e. topic locality on the Web [1,2]. Thus, the objective of the crawlers is to stay focused, that is, remaining within the neighborhood in which topic-specific pages have been identified. Focused crawlers work like general-purpose spiders, traversing the Web according to an appropriate traversal priority, instead of the Breadth-First or Depth-First ordering. The ideal focused crawlers retrieve the maximal set of relevant pages while simultaneously traversing the minimal number of irrelevant pages on the Web. This leads to significant savings in hardware and network resources, and helps keep the crawl more up-to-date.

The basic idea of a focused crawler is to optimize the visit priority of the candidate URLs in a crawling frontier that consists of URLs whose corresponding pages have yet to be fetched by the crawler. A URL should get a higher priority if the page pointed to by it has a higher relevant degree. In this paper, we introduce an innovative approach that combines four strategies to predict the relevance more effectively.

The main contributions of this paper are as follows.

(1) A cross-language hierarchical taxonomy is suggested to represent the topics. Based on the taxonomy users can select their interested topics in English or Chinese, and then the crawler can collect high relevant pages in English, Chinese or both. In addition, the topic context is used to weight a given topic and its contextual topics according to their relative hierarchies in the taxonomy.
(2) A formal description of the process of predicting the relevance of the uncrawled pages to a given topic is discussed.
(3) Four relevance predicting strategies based on page contents, anchor texts, URL addresses and link types of Web pages are introduced, respectively, to improve the relevance computation, in which, a special strategy evaluating the relevance based on the unique characteristic of the Chinese URLs is firstly proposed.
(4) A new focused crawling algorithm, named FCMRPS, is presented based on the combination of the above strategies and the Shark-Search algorithm [3], which estimates the relevance mainly based on the page content and anchor text.

Experiments were carried out on the Web for 30 topics in both English and Chinese. The experiments show that the FCMRPS can obtain significantly higher efficiency than these conventional crawling algorithms, i.e. Breadth-First, Best-First and Shark-Search.

The rest of this paper is organized as follows: Section 2 provides an overview of the focused crawling and the Shark-Search algorithm. Section 3 first introduces a cross-language taxonomy for topic description and then presents a formal description of the relevance predicting process and four relevance prediction strategies. Section 4 describes the details of FCMRPS. Section 5 shows some experimental results and discussions on focused search. Section 6 draws some conclusions and our future work.

2. Related work

Shark-Search [3] is a refined version of Fish-Search which is the first dynamic focused crawling algorithm. Fish-Search [4] is based on the schools of fish metaphor: A school of fish moves in the direction of food. Each URL corresponds to a fish whose survivability is dependent on visited page relevance and remote server speed. Page relevance is estimated based on the page textual content using a binary classification (the page can only be relevant or irrelevant). [5] presents an improved Fish-Search. It points out that the random of search range of original Fish-Search would lead to repeated search. Different fishes moving in different directions can be regarded as different directed graphs. A "distance" parameter that is the distance between the centers of two directed graphs is used to control the search direction. The distance is calculated in graph theory. So, by adjusting "distance" to be a reasonable value between different fishes adaptively, the repeated search problem can be solved. [3] extends Fish-Search into Shark-Search. The given topics are described in keywords. Two improvements are made to the original Fish-Search algorithm to overcome some limitations. One immediate improvement is that relevance between page content and topic is calculated by vector space model and can be any real number between 0 and 1. Another significant improvement is that candidate URLs to be downloaded are prioritized by taking into account a linear combination of page content and anchor text relevance on the source page. Experiments show that the Shark-Search performs between 1.5 and 3 times better than its ancestor does. In [6], a link analysis technology is used to improve the Shark-Search. Some literatures [7–13] make use of PageRank [14] as link analysis algorithm to evaluate the importance of candidate URLs. Although PageRank is effective to rank the results of search engines, they are not suitable for focused crawling for the reason that its process is computationally expensive and based on the overall Web graph [7,10–13]. Therefore, link type analysis is utilized to estimate the relevance of candidate URLs. Links are divided into five groups according to the relative position of the candidate URL to its parent in the Web graph. Then, five heuristic rules are presented to infer the topical relation of a page to its parent page

based on link types. According to these rules, different link relevance scores are assigned to the candidate URLs. The link relevance, page content relevance and anchor text relevance are combined to get the final relevance of the candidate URLs.

How to give a formal description for a given topic is the first step for focused crawling. Until now, topics are often represented in keywords [3], natural language text [15,16] and hierarchical taxonomy [10–12,17]. The first two models do not contain hierarchical context, but the last one does. [17] proposes an method to map the topic described in a keyword set or a text written in natural language to those described in hierarchical topic taxonomy. Then, a approach using the hierarchical topic context in the taxonomy is proposed to evaluate the relevance more effectively.

There has been much other research on relevance computation of focused crawling. In [18], two classifiers are used. The baseline classifier navigates through the Web to obtain enriching training data for the apprentice classifier. The apprentice classifier is trained over the data collected through the baseline classifier and determines the relevance only by the anchor context. Three kinds of crawling spiders, namely Breadth-First spider, PageRank spider and Hopfield Net spider are evaluated in [7]. Hopfield Net spider models the Web as a neural network in which the nodes are pages and the links are simply hypertext links. Links are weighted by the relevance of source page content and anchor text to the topic. The most weighted link is selected to fetch in every step. Experiments show that Hopfield Net spider outperforms the other two spiders.

Several strategies for prioritizing URLs based on the pages downloaded so far are proposed in [8]. Their strategies are based on considering some features such as PageRank or frequency of topic keywords in the page content to decide the priority of URLs to be crawled. They conclude that determining the priority based on PageRank value yields the best overall crawling performance. [19] proposes a focused crawler which is composed of three components, namely classifier, distiller and crawler. The classifier evaluates the relevance of page content to the topic to determine future link expansion. The distiller identifies those hub pages to determine the priorities of URLs to be visited. The crawling module fetches pages using the list of URLs provided by the distiller.

In [20], an ontology-based algorithm is used to compute relevance. After preprocessing, words occurring in the ontology are extracted from the page content. Relevance is then computed by using several measures on ontology graph (e.g. direct match, taxonomic and more complex relationships). A context graph is first built for seed pages using links to the pages returned from other search engine [21]. The context graph stores the information about the link hierarchies and the distances from the off-topic pages to the target pages which are used to determine the visiting priorities. A set of classifiers are trained based on the context graph to assign pages to different layers of the graph. However, it is infeasible to rely on other search engines to obtain backlink information. Furthermore, due to rapidly growing and mixing various topics in the Web graph, the assumption that all pages in a certain level from a target page will share terms does not always hold.

A collaborative geographically focused crawler is presented in [22], in which a group of crawling nodes is responsible for a specific portion of the Web, respectively. Several collaborative crawling strategies are proposed, whose goal is to collect Web pages about specified geographic locations, by considering features like URL address of page, content of page, extended anchor text of link, and others. Various evaluation criteria are used to qualify the performance of such crawling strategies and show that features like URL address of page and extended anchor text of link yield the best overall performance.

In general, three kinds of available features, i.e. page content, anchor text and link structure, are often used to predict the relevance. However, to the best of our knowledge, most of all existing focused crawlers evaluate the relevance only using one or two kinds of these features. In this paper, we utilize four kinds of known features, i.e. page content, anchor text, URL address and link type as well as the hierarchical topic context to improve relevance prediction and present a new focused crawling algorithm.

3. Relevance prediction

In this section, we introduce a hierarchical topic taxonomy to describe topics and formalize the process of relevance prediction. Four features including textual information, namely page content, anchor text and URL address, and link structure, i.e. link type, as well as the hierarchical topic context in the taxonomy are used to predict the relevance more precisely. In particular, a novel strategy based on the Chinese URLs address is presented to evaluate the relevance.

3.1. Cross-language topic description based on ODP

Focused crawlers are activated in response to particular information needs described as topics. These needs could be from an individual user (query time or online crawlers) or from a community with common interests (topical or vertical search engines and portals). Topics may be obtained from different sources as for instance asking users to specify them.

The "DMOZ" ODP (Open Directory Project) [23] is the largest, most comprehensive human-edited, hierarchical taxonomy currently available. It covers 4 million sites filed into more than 590,000 categories (16 wide-spread top-categories, such as Arts, News, Sports, etc.). ODP is organized as a tree where topics are internal nodes and examples of Web pages relevant to its parent topic node are the leaf nodes. ODP is first used in the directory navigating service such as Yahoo Directory Service [24] and Google Directory Service [25]. In addition, it is used in the personalized Web search to describe the user profiles [26,27]. As well as for focused crawling, ODP is used in [10–12,18,19,22]. ODP is regarded as the training data in [18, 19,22] to train a topical classifier which determines a downloaded page is either relevant or irrelevant to the topic. Topic keywords extracted from ODP are used to guide the crawler and evaluate the focused crawling in [10–12].

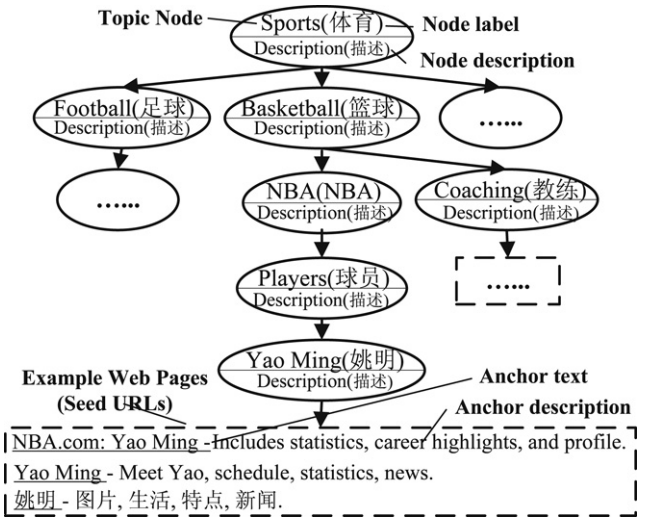

Fig. 1. Cross-language topics description example in ODP.

In this paper, we use ODP whose nodes are described in both English and Chinese, to describe the topics. Every internal node in ODP is a *topic node*, denoted by $Node_{label}$, where the subscript *label* is the name of the node. User can select one or more $Node_{label}$ as her interest topics. Once $Node_{label}$ is identified, *topic context path*, represented by $Path_{label}$, is the path from the root of ODP to $Node_{label}$ (including root and $Node_{label}$). Let $Depth_{label}$ denote the depth of the topic context path, namely the number of nodes in the $Path_{label}$. Then, the topic context path can be represented as $Path_{label} = \{label_1, label_2, \ldots, label_{Depth_{label}}\}$ in which $label_1$ is the label of ODP root and $label_{i(i=1\text{to}Depth_{label})}$ is the label of the ith $Node_{label}$.$label_{Depth_{label}}$ is the *Topic*, and the other $label_1, label_2, \ldots, label_{Depth_{label}-1}$ are its contextual topics. $Path_{label}$ is used for relevance prediction to guide the focused crawler. *Topic Subtree* is the subtree rooted at $Node_{label}$ and includes its all descendant nodes. All nodes description and anchor texts, anchor descriptions of these example Web pages are regarded as the *topic description* denoted by $Desc_{label}$. $Desc_{label}$ is used to fairly compute the relevance of crawled pages in following experiment.

Let $v_t = \{\{label_1, w_1\}, \{label_2, w_2\}, \ldots, \{label_{Depth_{label}}, w_{Depth_{label}}\}\}$ represent the vector of the given topic in which $\{label_i, w_i\}$ represents a term where $label_i$ is the term's name and w_i is the term's weight. It is obvious that v_t includes some hierarchical topic context information. For example, a hub page of topic "Sports" may contain some links pointing to pages related to the topic "NBA". Thus, *Topic* and its contextual topics are effectively weighted in terms of their relative hierarchies in ODP, i.e. $w_i = i/Depth_{label}$. The given *Topic*'s weight is $D_{label}/D_{label} = 1$. The weight of the contextual topic will decrease when the distance between the *Topic* and the contextual topic increases.

An example is illustrated in Fig. 1. Every internal node (marked as bold line ellipse) may be a *Topic*. In addition, there are many example Web pages (marked as dot line rectangle) under every topic. For a given *Topic* "NBA" which is corresponding to $Node_{NBA}$, $Depth_{NBA} = 3$, $Path_{NBA} = \{\text{Sports, Basketball, NBA}\}$ in which "Sports" and "Basketball" are regarded as the contextual topics, $Desc_{NBA} = \{\text{all nodes description, all anchor text and anchor description of the example Web pages of the topic subtree of "NBA"}\}$. Thereby $v_t = \{\{\text{Sports}, 1/3\}, \{\text{Basketball}, 2/3\}, \{\text{NBA}, 3/3\}\}$. Clearly, an outgoing URL in a page on topic "Basketball" will have a higher probability to reach the target page of "NBA" than that in a page on topic "Sports".

3.2. The formal description of relevance prediction

In this section, we formalize the process of relevance prediction. We classify the URLs according to their roles in the crawling process, as shown in Fig. 2, where *KUS* (Known URLs Set) represents all URLs that focused crawler has collected so far. The initial value of *KUS* is one or more seed URLs. Then *KUS* is classified into two groups: *CUS* (Crawled URLs Set) and *UCUS* (UnCrawled URLs Set). *CUS* is the set of all URLs that the pages pointed to by them have been fetched. *UCUS* is the set of all URLs that the pages pointed to by them have not been fetched. *UCUS* consists of all candidate URLs to be fetched in the crawling frontier. URLs will continue to be moved from *UCUS* to *CUS* through the crawling process. *CUS* is grouped into *RCUS* (Relevant Crawled URLs Set) and *ICUS* (Irrelevant Crawled URLs Set). *RCUS* is the URLs that the pages pointed to by them not only have been obtained but also are relevant to the topic. *ICUS* is the URLs that the pages pointed to by them have been obtained but are irrelevant to the topic. Finally, the *UCUS* is divided into two groups, namely *RUCUS* extracted from *RCUS* and *IUCUS* extracted from *ICUS*. In other words, *RCUS* and *ICUS* are the parents of *RUCUS* and *IUCUS*, respectively.

In the following, the predicting process will be simplified to evaluate the relevance of the pages pointed to by the URLs in *UCUS* to the topic in terms of the features seen so far from these four domains *RCUS, ICUS, RUCUS* and *IUCUS*. The candidate URLs in *UCUS* are prioritized by taking into account a linear combination of four features, namely source page content, anchor text, URL address and link type.

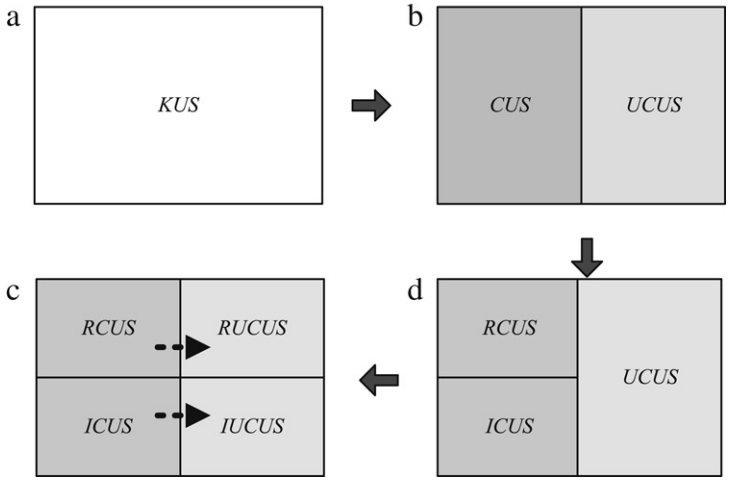

Fig. 2. Classification of Known URLs.

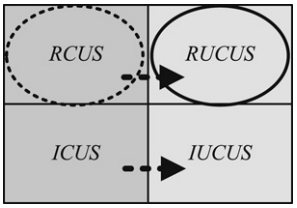

Fig. 3. Relevance prediction based on page content.

In addition, we define some symbols. Let *Topic* represent the user interested topic. Let $url^{current} \in \{CUS\}$ be the current crawling URL and point to the page $page^{current}$ that has been fetched. $content^{current}$ is the textual content of $page^{current}$. $url^{child} \in \{UCUS\}$ is the unvisited URL of an outgoing link on $page^{current}$ and points to $page^{child}$ that has not been crawled. $anchor^{child}$ represents the anchor text of url^{child} and $anchor\_context^{child}$, which is extracted between the given predefined boundaries, is the textual context of $anchor^{child}$. The link type of url^{child} is url^{type}, which will be discussed further in Section 3.6. Then, the quadruple $\{content^{current}, \{anchor^{child}, anchor\_context^{child}\}, url^{child}, url^{type}\}$ is used to predict the relevance of $page^{child}$ to the *Topic*.

3.3. Relevance prediction based on page content

Page content is the base of focused crawling used to evaluate the relevance. Let $url^{current} \in \{RCUS\}$ and $url^{child} \in \{RUCUS\}$ be the source and target, respectively, as shown in Fig. 3. The underlying assumption of relevance prediction based on page content is that the target page inherits the relevance of the source page. Therefore, the page content relevance prediction, denoted by R_{PC}, is that $content^{current}$ is utilized to evaluate the relevance of $page^{child}$ to the topic. R_{PC} is calculated as the cosine similarity between $content^{current}$ and *Topic* in function (1).

$$R_{PC} = \frac{v_c \bullet v_t}{|v_c| \times |v_t|} \tag{1}$$

where v_c is the vector of $content^{current}$ represented by TF (Term Frequency) [28] after removing stop words and stemming.

3.4. Relevance prediction based on anchor text

Fig. 4 describes the relevance prediction based on anchor text. A URL's anchor text and context are good indicators of its target page content. Let $url^{current} \in \{RCUS\}$ and $url^{child} \in \{RUCUS\}$. The relevance prediction is that the anchor text $anchor^{child}$ and anchor context $anchor\_context^{child}$ in $page^{current}$ is used to estimate the relevance of $page^{child}$ to the topic represented by R_{anchor} and $R_{anchor\_context}$, respectively.

$$R_{anchor} = \frac{v_{anchor} \bullet v_t}{|v_{anchor}| \times |v_t|} \tag{2}$$

$$R_{anchor\_context} = \frac{v_{anchor\_context} \bullet v_t}{|v_{anchor\_context}| \times |v_t|} \tag{3}$$

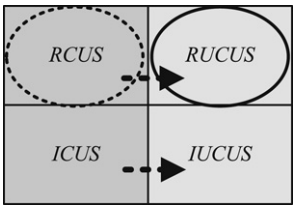

Fig. 4. Relevance prediction based on anchor text.

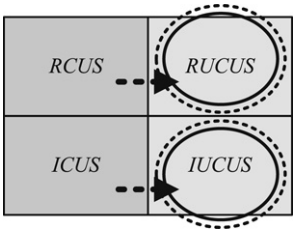

Fig. 5. Relevance prediction based on URL address.

v_{anchor} and $v_{anchor\_context}$ are the vectors of $anchor^{child}$ and $anchor\_context^{child}$ represented by TF after removing stop words and stemming, respectively. Hence, the combining relevance R_{AT} of R_{anchor} and $R_{anchor\_context}$ is as follows.

$$R_{AT} = \begin{cases} R_{anchor} & \text{if } R_{anchor} > 0 \\ R_{anchor\_context} & \text{if } R_{anchor} = 0. \end{cases} \tag{4}$$

3.5. Relevance prediction based on URL address

Let $url^{child} \in \{UCUS\}$, the URL relevance prediction based on URL address, as shown in Fig. 5, is that the tokens contained in url^{child} is utilized to estimate the relevance of $page^{child}$ to the topic, represented by R_{url}. Web sites are often organized by directories (i.e. topics) [2]. Most URLs are each manually assigned a directory regardless of whether they are automatically generated by using some programs or not. For example, when a Web site master publishes a page about "NBA", she will assign the page to the directory of "sports" (more specific "Basketball"). And the URL of the page should contain "sports" ("basketball"), such as "http://sports.sohu.com/2008/20080229.htm". Therefore, URLs of most Web pages are associated semantic meanings with the pages content. We assume that the tokens in a URL can be used to predict the relevance of the page pointed to by this URL to the topic. This idea is experimentally confirmed in the following Section 5.3.2. The URLs of English Web pages are mostly composed of some meaningful English words and have been used in focused crawling [17, 29]. However, the URLs of Chinese Web pages have their own characteristics. We classify the most tokens of Chinese URLs into two groups.

Group 1 : A Token is composed of English Words. It is the same with that of English URLs. In general, the English Word appears in two forms: full word or abbreviation. For example, "Sports"("体育" in Chinese) in the URL "http://www.chinadaily.com.cn/sports" and "info" (the abbreviation of "Information" and "信息" in Chinese) in the URL "http://info.tsinghua.edu.cn".

Group 2 : A Token is composed of Chinese Pinyin. In most cases, the Chinese Pinyin also appears in two fashions: full Pinyin or the first letters of Pinyin. For example, the Chinese Pinyin "TiYu" of "体育" ("Sports" in English) in the URL "http://www.dzwww.com/tiyu" and the first letters "sh" of "上海" whose Chinese Pinyin is "ShangHai" in the URL "http://www.sh.xinhuanet.com".

Additionally, in most cases, English words or Chinese Pinyin are concatenated in two ways: (a) direct concatenation, such as the "YaoMing" of "姚明" in "http://www.yaoming.net"; (b) concatenated by a under line "_", such as "Tai_Gang_Ao" of "港澳台" in "http://www.xinhuanet.com/tai_gang_ao". According to above rules, Our system first automatically parses 3,570,000 Chinese URLs extracted from the CWT200g [30] into tokens in terms of "/" and "." in URLs as the separators. Then, for each token with frequency more than 500, we manually associate it with a topic. Finally, we get a Topic-Token Mapping Table (TTMP) which map 556 most commonly used topics to corresponding tokens. Example is shown in Table 1. Furthermore, we build a feedback mechanism to recognize new tokens in crawling process. When a new token is not in the TTMP and its appearance times is greater than the threshold 500, it will be manually mapped to a topic and added to the TTMP.

In the following, we will discuss the URL relevance prediction in detail. For url^{child}, we first delete the string "http://". The remainder of the url^{child} is parsed into $block_{j(j=1 \text{ to } J)}$ in term of the "/" characters in the URL in which J is the

Table 1
Example of topic-token mapping table.

Topic		English token		Chinese pinyin token	
English	Chinese	Full	Abbreviation	Full	First letters
Sports	体育	sports		tiyu	
Software	软件	software	soft	ruanjian	
Automobile Car	汽车	automobile car	auto	qiche	
ShangHai	上海	shanghai		shanghai	sh

number of blocks in the url^{child}. Then for every $block_j$, it is parsed into $token_{jk(k=1\text{ to }K)}$ according to "." where K means there are K tokens in the $block_j$. In addition, we filter out all stop tokens such as "com", "html" and so on. For $label_i$ in $v_t = \{\{label_1, w_1\}, \{label_2, w_2\}, \ldots, \{label_{Depth_{label}}, w_{Depth_{label}}\}\}$, if it exists in TTMP, we first represent it by $label_{iq\,(q=1\text{ to }4)} = \{label_{i1}, label_{i2}, label_{i3}, label_{i4}\}$ in which $label_{i1}$, $label_{i2}$, $label_{i3}$ and $label_{i4}$ are the tokens of full English, abbreviated English, full Chinese Pinyin and first letters of Chinese Pinyin, respectively. Notice that $label_{iq}$ should be "NULL" if there is no corresponding token in TTMP. If $label_i$ does not exist in TTMP, we represent it in {$label_i$, NULL, Full Pinyin of $label_i$, NULL} in which "Full Pinyin of $label_i$" is obtained in terms of a Chinese Pinyin Dictionary.

For $label_{iq(i=1\text{ to }Depth_{label};q=1\text{ to }4)}$, let B denote the label-to-token matching matrix.

$$B = b_{iq} = \begin{cases} \alpha_q & \text{if } \exists j, k;\ label_{iq} \in token_{jk} \\ 0 & \text{if } label_{iq} = \text{NULL or if } \forall j, k;\ label_{iq} \notin token_{jk} \end{cases} \tag{5}$$

in which B is a matrix with dimension $Depth_{label} \times 4$, b_{iq} is the strength of the match between $label_{iq}$ and $token_{jk}$, and α_q is a predefined constant. There may be an occasional mistake during the matching process, such as that "sh" of "ShangHai (上海)" matches with "shop". Moreover, the error probability of $label_{i4}$ and $label_{i2}$ is higher than that of $label_{i1}$ and $label_{i3}$. Consequently we set $\alpha_1 = \alpha_3 = 1$, $\alpha_2 = 0.5$ and $\alpha_4 = 0.2$. Note that it is not case sensitive in the match process.

Finally, the relevance score R_{url} is as follows where $\text{Max}\{b_{iq\,(q=1\text{ to }4)}\}$ denotes the maximum of b_{iq} and $\beta = 1/(\sum_{i=1\text{ to }Depth_{label}} \frac{i}{Depth_{label}})$ used to normalize R_{url} to a value between 0 and 1.

$$R_{url} = \sum_{i=1\text{ to }Depth_{label}} w_i * \text{Max}\{b_{iq(q=1\text{ to }4)}\} * \beta. \tag{6}$$

The URL relevance predicting Algorithm 1 is as follows. To get a better understanding for this, an example is given based on the ODP in Fig. 1. Assume that the query topic is "篮球 " ("Basketball" in English and "LanQiu" in Chinese Pinyin), thus $v_t = \{\{体育(\text{Sports}), 1/2\}, \{篮球(\text{Basketball}), 2/2\}\}$. Then we get $label_{iq(i=1\text{ to }2,q=1\text{ to }4)} = \{\{label_{11}, \ldots, label_{14}\}, \{label_{21}, \ldots, label_{24}\}\} = \{\{\text{"sports", NULL, "tiyu", NULL}\}, \{\text{"basketball", NULL, "lanqiu", NULL}\}\}$ according to TTMP. For url^{child} = "http://sports.sohu.com/lanqiu.shtml", it is first parsed into $token_{jk(j=1\text{ to }2)} = \{token_{12}, token_{21}\} = \{\{\text{sports, sohu}\}, \{\text{lanqiu}\}\}$. It is obvious that "sports" and "lanqiu" can be matched. Consequently $b[2][4] = \{\{1, 0, 0, 0\}, \{0, 0, 1, 0\}\}$. Therefore, the relevance score $R_{url} = (1/2 * 1 + 2/2 * 1) * 2/3 = 1$.

Input: URL url^{child}, topic v_t
Output: link relevance R_{url}
1 delete "http://" from the url^{child};
2 parse the remainder of the url^{child} into blocks $block_{j\,(j=1\text{ to }J)}$ according to "/";
3 for j = 1 to J do parse $block_j$ into tokens $token_{jk\,(k=1\text{ to }K)}$ according to ".";
4 B= b[$Depth_{label}$][4]=0; //initialize the matrix
5 for i = 1 to $Depth_{label}$ do
6 get $label_{iq\,(q=1\text{ to }4)}$ in terms of TTMP;
7 compute B according to the function 5;
8 $R_{url} = R_{url} + w_i * \text{Max}\{b_{iq}\} * \beta$;
9 return R_{url};

Algorithm 1: URL Relevance Predicting Algorithm

3.6. Relevance prediction based on link structure

In Fig. 6, the link relations are built from the source $url^{current} \in \{RCUS\}$ to the target $url^{child} \in \{RUCUS\}$. In [6], we have proposed using link analysis technology to improve the Shark-Search. We classify the links into five types represented by url_{type}, i.e. downward, sibling, crosswise, outward and upward, in terms of the relative locations of $url^{current}$ and url^{child} in the Web graph. Then five heuristic rules are presented to infer the topical relation of $page^{child}$ to its parent $page^{current}$. That is, if

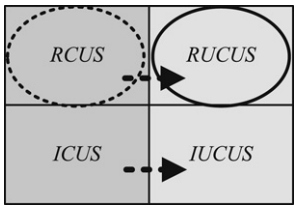

Fig. 6. Relevance prediction based on link structure.

the url_{type} is downward then $page^{child}$'s topic is a specialization of $page^{current}$'s topic and a higher relevance score is assigned to url^{child}. If $page^{current}$ is a content page and the url_{type} is sibling then $page^{child}$ is on the same topic with $page^{current}$ and a high relevance score is assigned to url^{child}. If the url_{type} is crosswise then $page^{child}$ is likely to be on the same topic with $page^{current}$ and a high relevance score is assigned to url^{child}. If the url_{type} is outward then $page^{child}$ is likely to be on the same topic with $page^{current}$ and a middle relevance score is assigned to url^{child}. If the url_{type} is upward then $page^{child}$ is likely to lead to topics that are more general and a low relevance score is assigned to url^{child}. Different link types have various impacts on the candidate URLs priority, therefore we assign different relevance to the url^{child} according to its link type.

$$R_{LT} = \begin{cases} \gamma_1 & \text{if } R_{AT} > 0 \text{ and } R_{PC} > 0 \\ \gamma_2 & \text{if } R_{AT} = 0 \text{ and } R_{PC} > 0 \\ 0 & \text{if } R_{AT} = 0 \text{ and } R_{PC} = 0. \end{cases} \tag{7}$$

In which γ_1 and γ_2 ($\gamma_1 > \gamma_2$) are predefined constants, and γ_2 is assigned according to the link type. The $\gamma_1 > \gamma_2$ just ensure that the priority of R_{AT} is superior to that of R_{LT}. The optimized γ_1 and γ_2 are found in experiment that maximizes the relevant pages of the search result.

3.7. Relevance combination

The combining relevance score *RS* may be a weighted product of each individual relevance scores. Equivalently, we can get the aggregate relevance score by summing the weighted individual relevance scores.

$$RS = w_1 * R_{PC} + w_2 * R_{AT} + w_3 * R_{url} + w_4 * R_{LT}. \tag{8}$$

Here w_1, w_2, w_3 and w_4 are weights used to normalize the different relevance factors. By increasing the weight of a given factor, we can increase the importance of the corresponding individual. In our particular implementation, we chose to use weights such that individual relevance score was almost equally balanced. We will see that one of the interesting outcomes of such a strategy is that even though the different factors perform differently depending upon the nature of the predict and the starting seed, the over all performance of the combination of more than one factors was almost superior to each individual factor. We will provide further insights on this issue in the experimental section.

4. Focused crawling algorithm

This section presents the FCMRPS algorithm based on the relevance prediction methods discussed in Section 3 and the Shark-Search, as shown in Algorithm 2.

Input: url^{seed}, depth (D), the number (N) of the pages to be crawled and a given topic v_t
Output: page set *pagesSet* related to the topic
1 url^{seed}.depth = D; $url^{seed} \Rightarrow UF$ (URLs priority Frontier);
2 while UF is not empty and $pagesSet$.size $< N$
3 remove the first element of UF to $url^{current}$;
4 crawl $page^{current}$; $page^{current} \Rightarrow pagesSet$;
5 for each uncrawled outgoing link url^{child} in $page^{current}$;
6 compute R_{PC}, R_{AT}, R_{url}, R_{LT} and RS in terms of function (1), (4), (6), (7) and (8), respectively;
7 if $R_{PC} > \delta$ then set url^{child}.depth = D where δ is a predefined threshold constant; else url^{child}.depth = $url^{current}$.depth - 1;
8 if url^{child} already exists in UF then set $url^{child}.RS$ = Max{the existing RS in UF, the new RS} and reorder UF in terms of RS if necessary and url^{child}.depth = Max{the existing depth in UF, the new depth};
9 else if url^{child}.depth > 0 then insert url^{child} at its right location in UF;
10 endwhile;
11 return $pagesSet$;

Algorithm 2: the FCMRPS Algorithm

This algorithm allows users to dynamically search sub-areas of the Web predefined by a given "hop" parameter (i.e. depth (D) in the Web graph). The key principle of the algorithm is the following: it takes as input some seed URLs url^{seed} and a search topic v_t, and dynamically builds a URLs frontier UF (initialized to the seed URLs and prioritized by the Relevance Score RS). At each step the first element $url^{current}$ is popped from the frontier and processed. As the content of page $page^{current}$ is downloaded, it is analyzed by these four strategies evaluating the relevance score to the search topic. URLs in UF are prioritized in terms of RS. Each url^{child} is assigned a depth value. If $page^{current}$ is relevant, the depth of url^{child} is set to the predefined value D. Otherwise, the depth of url^{child} is set to be one less than the depth of $url^{current}$. When the depth reaches zero, the direction is dropped and none of its outgoing links is inserted into the UF. url^{child} whose depth is greater than 0 is inserted at the right location in UF according its RS. The algorithm runs until it has collected the predetermined number N of pages.

5. Experiments

We compared the efficiency of our FCMRPS algorithm to three other standard crawling algorithms: (1) the Breadth-First algorithm; (2) the Best-First algorithm; (3) the Shark-Search algorithm. Experiments were conducted on the Web for different English and Chinese topics and Web sites. Our experiments show that the FCMRPS algorithm outperforms the other algorithms significantly.

5.1. Evaluation measures

There are three measures that may be used to evaluate the performance of a crawler: recall rate, precision rate (harvest rate) and sum of information [3,10–13]. The first two measures are traditional information retrieval criteria for evaluating effectiveness of search engines. Recall is the ration of the number of relevant documents retrieved to the total number of relevant documents in the range retrieved. However, recall cannot be evaluated on large portions of the Web, because it would be too difficult to build a standard "test site" with enough coverage to evaluate the quality of results. So recall is not suitable for focused crawling. In this paper, we studied precision rate and sum of information in detail.

In order to fairly compare these algorithms, we use a new function (9) to compute the relevance of crawled pages to the given topic, in which v_c and v_t' are the vectors of $content^{current}$ and topic description $Desc_{label}$, respectively.

$$R'_{PC} = \frac{v_c \bullet v_t'}{|v_c| \times |v_t'|}. \tag{9}$$

Precision (harvest) rate measures the query result at page level. To simplify the evaluation, we therefore introduce a simple notion of "relevant page", that a page is a relevant page if its R'_{PC} is greater than a certain threshold. So,

$$precision\_rate = \frac{n_1}{N} \tag{10}$$

where n_1 is the number of relevant pages retrieved, N is the number of all pages retrieved.

The sum of information evaluates the result regarding all collected pages as a whole. We define R'_{PC} as the sum of information of a page to the topic. Thus, the sum of information of the complete retrieved pages set $pagesSet$ is as follows:

$$sum\_of\_info = \sum_{(\text{every page in } pagesSet)} R'_{PC}. \tag{11}$$

Our measures verify the result from different granularity and satisfyingly reflect what most users expect from a "good" search engine: getting as many relevant pages in the shortest delays.

5.2. Simulation

5.2.1. Baseline crawling algorithms

We ran these crawlers on the real Web and compared the effectiveness of our FCMRPS algorithm to the other three baseline crawling algorithms.

(1) Breadth-First Algorithm

The Breadth-First algorithm is the simplest strategy for crawling. This algorithm was explored as early as 1994 in the Web crawler [31] as well as in more recent research [32]. It uses the frontier as a FIFO (First In First Out) queue, crawling URLs in the order in which it encounters them. Breadth-First is used here as a baseline crawler; since it does not use any feature to predict the relevance, we expect its performance to provide a lower bound for any of the other algorithms.

(2) Best-First Algorithm

Best-First crawlers have been studied in [8]. The basic idea is that given a frontier of URLs, the best URL according to some estimation criteria is selected for crawling. In our "naive" implementation, the URLs selection process is guided by simply computing the relevance between the topic and the page content for the URL, i.e. R_{PC} of this paper. Thus, $RS = R_{PC}$ is used to estimate the relevance of url^{child}. The URL with the best R_{PC} is then selected for crawling in every step.

Table 2
Examples of topics and seed URLs.

Topic		Seed URLs
English	Chinese	
Sports		sports.yahoo.com www.chinadaily.com.cn cbs.sportsline.com
	体育	sports.china.com cn.sports.yahoo.com sports.sina.com.cn
Basketball		www.chinadaily.com.cn/sports www.jes-basketball.com www.fandraftbasketball.com
	篮球	cn.sports.yahoo.com/basketball sports.sohu.com/lanqiu.shtml sports.sina.com.cn/basketball

Table 3
Average *precision_rate* and *sum_of_info* for 30 topics.

	Breadth-First	Best-First	Shark-Search	FCMRPS
precision_rate	2.72%	7.02%	6.05%	29.67%
sum_of_info	27.44	81.59	71.36	750.58

(3) Shark-Search Algorithm

Shark-Search can be seen as a variant of Best-First, with a more sophisticated relevance evaluation function. Shark-Search uses the similarities of page content and anchor text to the topic as the relevance, i.e. R_{PC} and R_{AT} of this paper, respectively. In [3], it was pointed out that when $RS = 0.2 * R_{PC} + 0.8 * R_{AT,}$ the crawler can get the best result.

5.2.2. Topics and seed URLs

In order to evaluate the crawling algorithms, we need topics and some corresponding seed URLs. We collected and extended 30 topics in both Chinese and English with different specificity (e.g. that "NBA" is more specific than "Sports") from ODP. Table 2 shows a few sample topics and seed URLs.

5.2.3. Parameters selection

At run time, the value of depth D was always set to 3 which is the same with that of the Shark-Search. For the sakes of limited bandwidth, memory and computational resources, the number N to be crawled pages was 10,000. For α, we let $\alpha_1 = \alpha_3 = 1$, $\alpha_2 = 0.5$ and $\alpha_4 = 0.2$. If the url_{type} is *downward*, $\gamma_1 = 0.4$, $\gamma_2 = 0.2$; if url_{type} is *sibling*, $\gamma_1 = 0.2$, $\gamma_2 = 0.1$; if url_{type} is *crosswise*, $\gamma_1 = \gamma_2 = 0.1$; if url_{type} is *outward*, $\gamma_1 = 0.2$, $\gamma_2 = 0.01$; and if url_{type} is *upward*, $\gamma_1 = 0.2$, $\gamma_2 = 0$. In addition, we set $\delta = 0.1$.

5.3. Results and discussion

5.3.1. General performance

Table 3 shows the average precision rates and sums of information of these four algorithms. For this case, we set $w_1 = w_2 = w_3 = w_4 = 1/4$ in the FCMRPS algorithm. It is obvious that the FCMRPS outperforms the other three algorithms significantly because it combines four strategies to predict the relevance more precisely. As expected, the Breadth-First is the least effective. To our surprise, the performance of Shark-Search is slightly less effective than Best-First. The main reason for this may be that few URLs' anchor texts are relevant to the topic explicitly.

To analyze the data further, we studied their dynamic performance during different stages of the crawling process. We divided the total 10,000 pages into 20 equal portions, each containing 500 consecutive pages according to the original visiting order of each crawler. Within each portion, we calculated the number of relevant pages and sum of information of the search results.

Fig. 7 illustrates the total number of relevant pages and Fig. 8 describes the total *sum_of_info*. We see that the performance of the FCMRPS is always superior to the other algorithms during the over crawling process. The performance of Best-First and Shark-Search are almost equally and they both outperform the Breadth-First. Breadth-First displays the worst performance and provides us with a baseline for all algorithms. Note that the Breadth-First, Best-First and Shark-Search have converged after about 8000 pages and the FCMRPS also tends to converge after about 10,000 pages, therefore it is not necessary to learn much more from running even longer crawls.

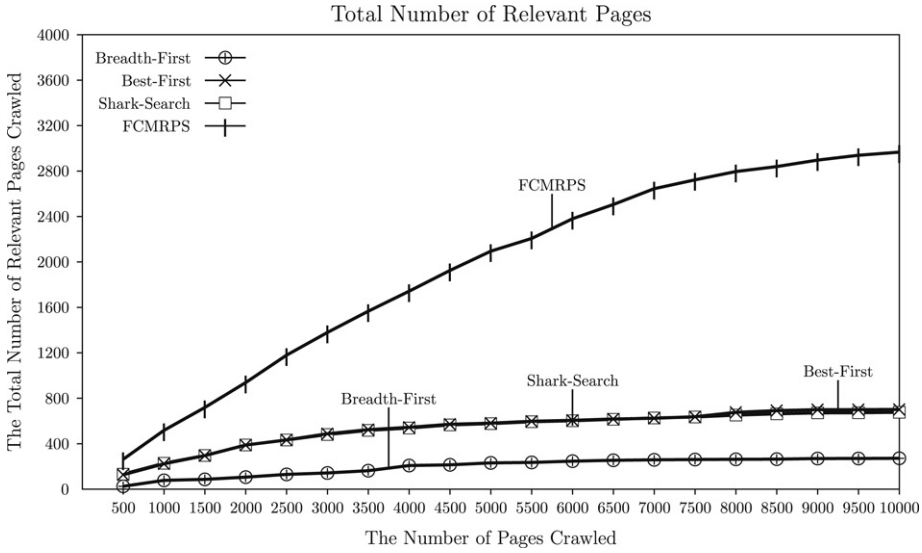

Fig. 7. The total number of relevant pages crawled for 30 topics in both English and Chinese.

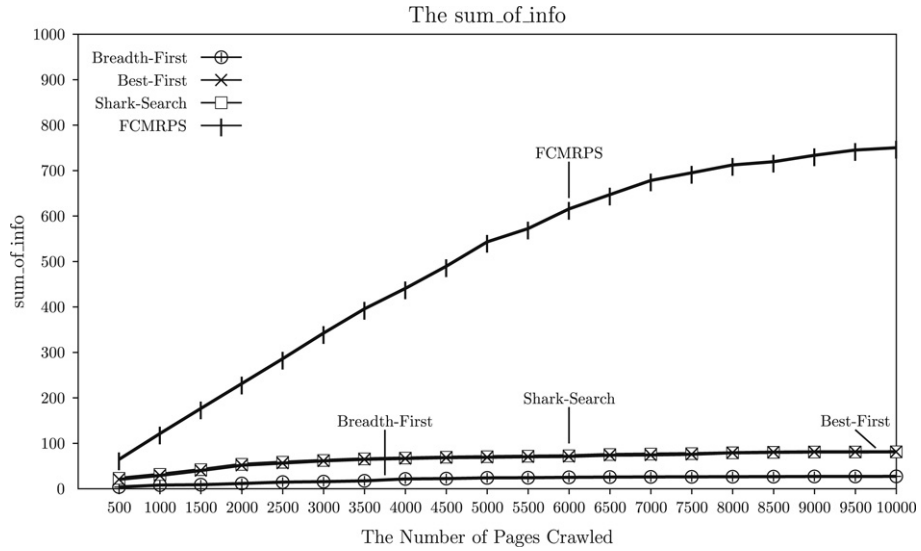

Fig. 8. The total *sum_of_info* for 30 topics in both English and Chinese.

5.3.2. URL relevance performance

In this section, we first verify the idea of relevance prediction based on URL address by analyzing whether the URLs of relevant pages in *pagesSet* contain the corresponding topic token or contextual topic tokens. Let Ra_t denote the ratio of the number of relevant pages whose URLs contain topic token to the total number of relevant pages. Ra_{ct} represents the ratio of the number of relevant pages whose URLs contain at least one contextual topic token to the total number of relevant pages. Ra is the sum of Ra_t and Ra_{ct}. We calculated Ra_t, Ra_{ct} and Ra for above four algorithms, respectively, as shown in Table 4. Although all Ra_ts are low, the Ra_{ct}s are very high. It means that most pages are associated with a certain topic (especially the top contextual topic) and their URL addresses contain the corresponding topic tokens. Therefore, tokens of a URL address may be used to predict the content of the URL's target page. In addition, the results in Table 4 agree with the general performance in Section 5.3.1. The FCMRPS get the highest values for Ra_t, Ra_{ct} and Ra among the four. The Best-First and Shark-Search obtain similar performance and both marginally better than the Breadth-First.

Then, we evaluate the effectiveness of URL relevance prediction based on URL address. We set $w_1 = w_3 = 1/2$, $w_2 = w_4 = 0$ (denoted by FCMRPS_{13}) and compared the dynamic performance of the FCMRPS_{13} and the Best-First in English and Chinese, respectively. The total number of relevant pages and sum of information are shown in Figs. 9–12, respectively. It is obvious that FCMRPS_{13} surpasses the Best-First significantly throughout the process. Therefore, the strategy of URL relevance prediction based on URL address is efficient for English especially for Chinese.

Table 4
Ratios of URLs containing topic or contextual topics tokens.

	Breadth-First	Best-First	Shark-Search	FCMRPS	Average
Ra_t	4.16%	6.58%	5.78%	9.92%	6.61%
Ra_{ct}	75.48%	73.66%	69.84%	86.39%	76.34%
Ra	79.64%	80.24%	75.62%	96.31%	82.95%

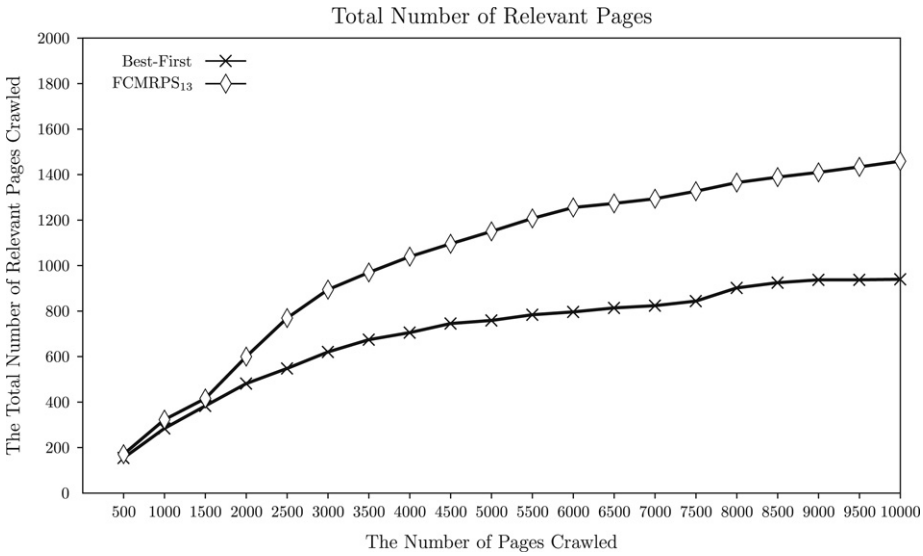

Fig. 9. The total number of relevant pages for English Topics.

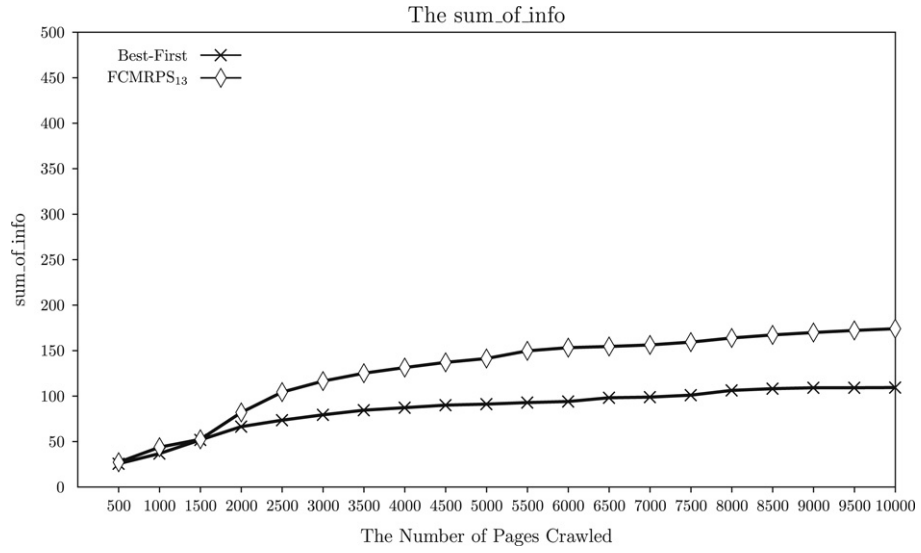

Fig. 10. The total *sum_of_info* for English Topics.

Furthermore, we set $w_1 = w_2 = w_4 = 1/3$ and $w_3 = 0$, denoted by FCMRPS_{124}. We studied the dynamic performance of Best-First, FCMRPS_{124} and FCMRPS, as depicted in Figs. 13 and 14. FCMRPS, FCMRPS_{124} and Best-First make use of four strategies, three strategies and one strategy, respectively, to guide the crawling process. It is desirable that an algorithm with more strategies will predict the relevance more accurately. As expected, FCMRPS_{124} performed significantly below FCMRPS, but outperformed Best-First.

5.3.3. Individual performance

As described in Section 3.6, the strategy of relevance prediction based on link structure cannot be separately used to guide the crawler because it depends on R_{PC} and R_{AT}. So we compared the performance of the other three individual relevance

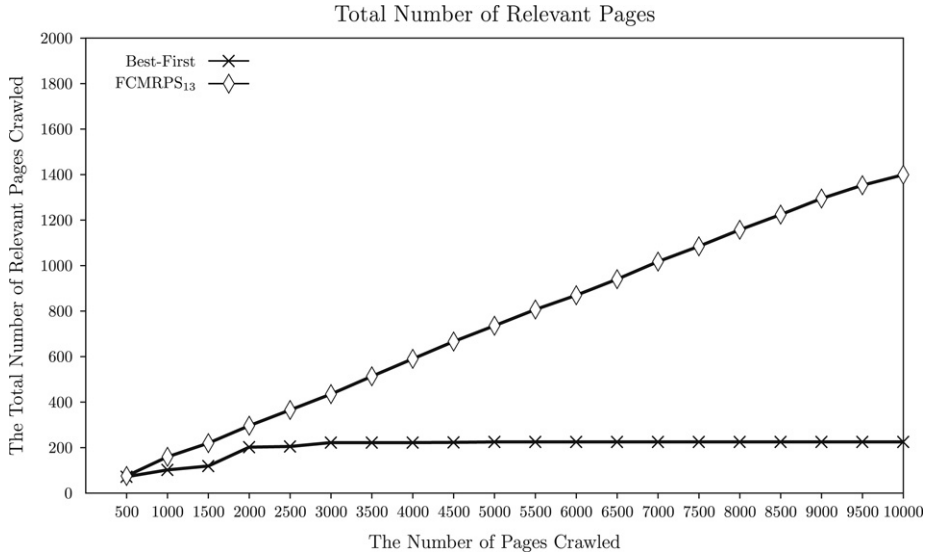

Fig. 11. The total number of relevant pages for Chinese Topics.

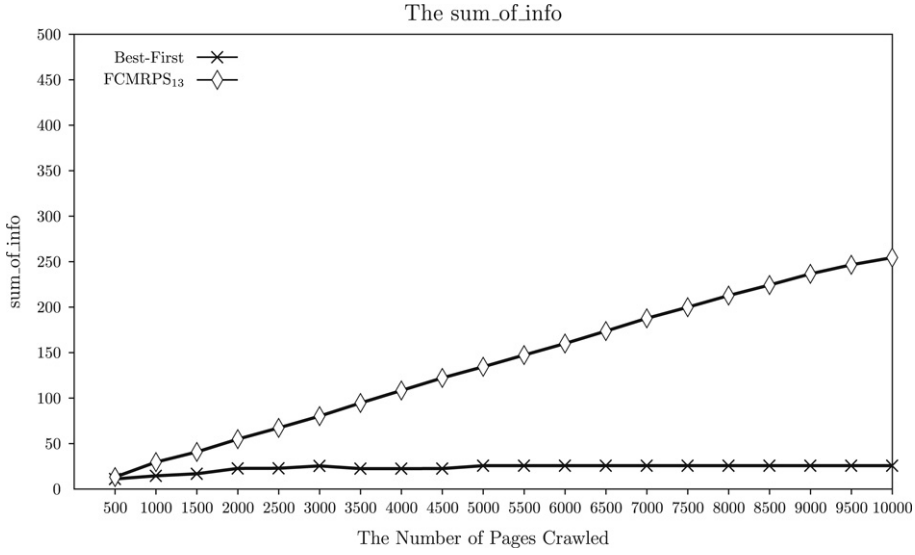

Fig. 12. The total *sum_of_info* for Chinese Topics.

strategies. Let $FCMRPS_1$ (i.e. Best-First), $FCMRPS_2$ and $FCMRPS_3$ represent the strategies of relevance prediction based on Page Content, Anchor Text and URL Address, respectively. Figs. 15 and 16 illustrate the experiment result. Pages about the same topic tend to be linked together and a page's content has rich information to be used to predict the relevance of its out links. Consequently, the $FCMRPS_1$ achieved the best performance. The $FCMRPS_3$ was slightly less effective. Most URLs contain contextual topic tokens which can be used to predict the relevance of their target pages. For example, the page pointed to by "http://sports.sohu.com/20080305/n255534156.shtml" will have a higher probability on topic "NBA" than that pointed to by "http://auto.sohu.com/20080305/n255526265.shtml". The $FCMRPS_2$ performed at a considerably lower level. The main reason for this may be that there are few URLs whose anchor texts are explicitly relevant to the topic.

5.3.4. Time performance

We discuss the time performance of above four algorithms in this section. The crawlers ran over a PC with Intel Core 2.33 GHz Dual processors, 2 GB of RAM. The running time of the crawling process consists of the following two core components: (1) *Downloading time* — the time of downloading the given number of Web pages; (2) *Processing time* — the running time of all the other operations (such as page content parsing, links extraction and relevance computation, etc.). We tracked the average running time of crawling 10,000 pages, as shown in Table 5. It is unnecessary to compare *Downloading time*s between these algorithms since it is impossible to control the impact of network traffic and congestion. The *Processing time* of Breadth-First

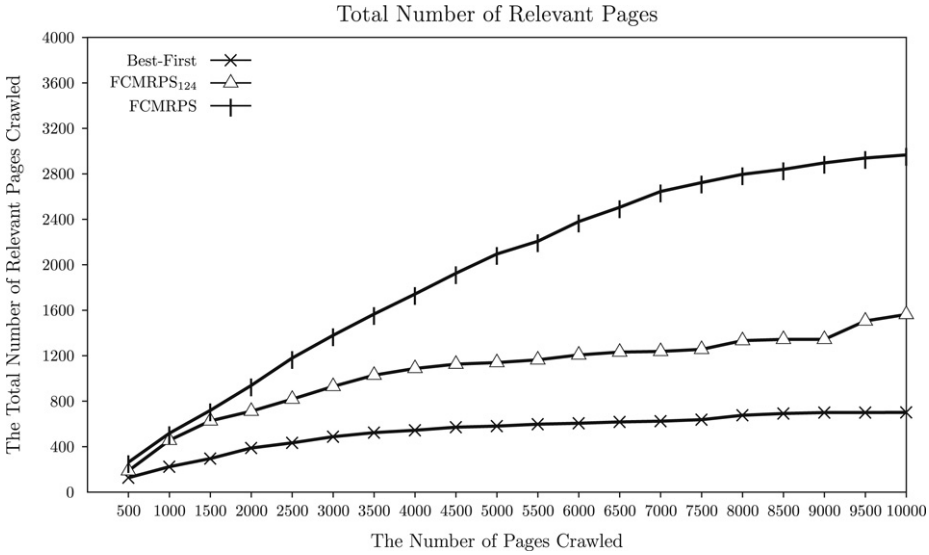

Fig. 13. The total number of relevant pages for Best-First, $FCMRPS_{124}$ and FCMRPS.

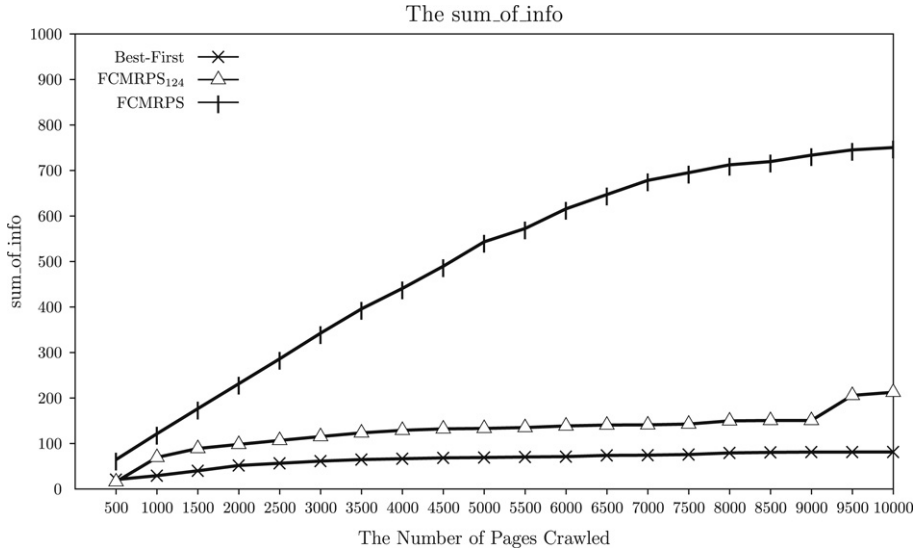

Fig. 14. The total *sum_of_info* for Best-First, $FCMRPS_{124}$ and FCMRPS.

Table 5
Average running time (minutes) of crawling 10,000 pages.

	Breadth-First	Best-First	Shark-Search	FCMRPS
Downloading time	35.65	31.93	38.63	36.01
Processing time	2.26	2.34	2.51	2.60
All Running time	37.91	34.27	41.14	38.61

is the shortest of 2.26 min among these algorithms. The *Processing time*s of Best-First and Shark-Search are 2.34 and 2.51 min, respectively. And that of FCMRPS is the longest of 2.60 min. Obviously the more complex the crawling algorithm, the longer the *Processing time*. However, the *Processing time* differences between these algorithms are rather negligible. In addition, compared with *Downloading time*, the *Processing time* can be ignored completely for the overall crawling process. Therefore, the running time of FCMRPS is almost the same as those of other three algorithms. However, FCMRPS can collect more relevant pages earlier.

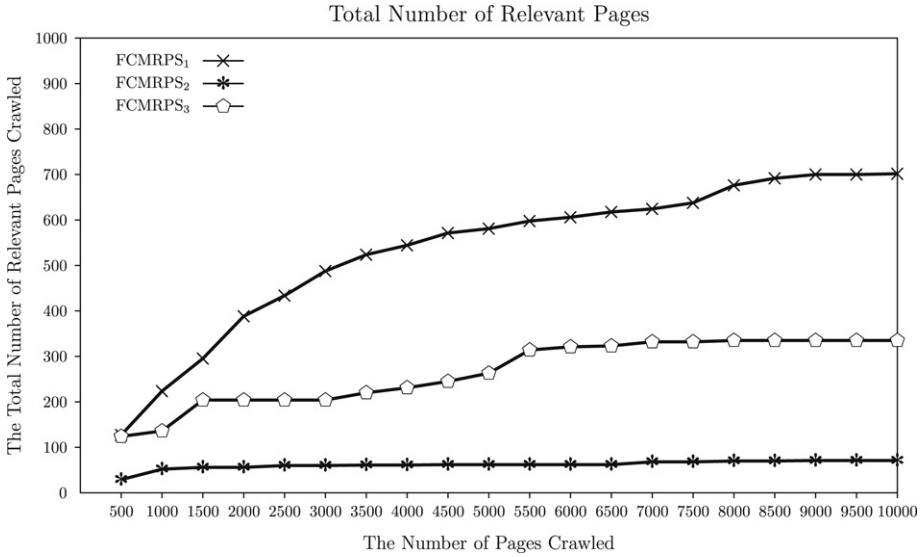

Fig. 15. The total number of relevant pages for $FCMRPS_1$, $FCMRPS_2$ and $FCMRPS_3$.

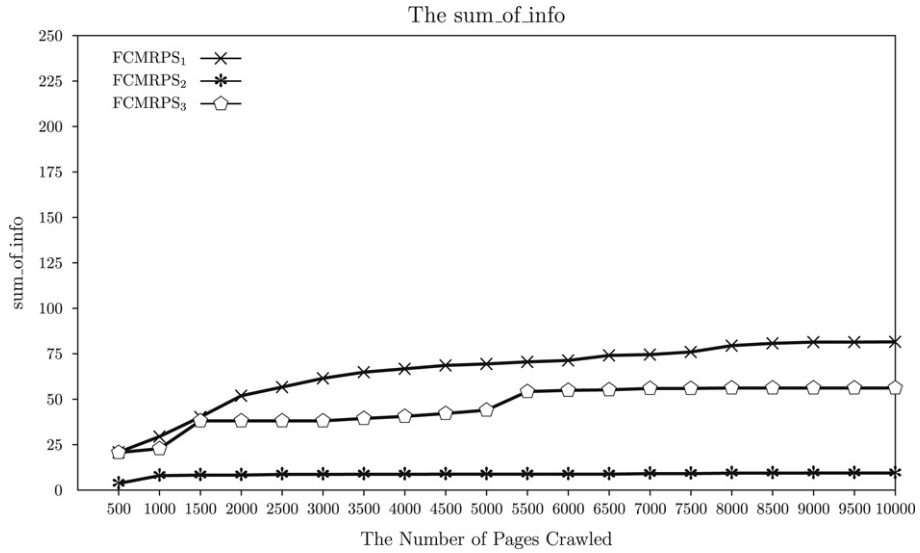

Fig. 16. The total *sum_of_info* for $FCMRPS_1$, $FCMRPS_2$ and $FCMRPS_3$.

6. Conclusions and future work

In this paper, we study how to predict the relevance of unvisited pages in focused crawling. We start at the description of users' queries, then propose a novel approach for the relevance predicting by combining four predicting strategies based on the features of the current Web pages, where the features include the page contents, anchor texts, URL addresses and link types. The prediction is suitable for the Web pages written in Chinese, English or both. We use this method to get a novel focused crawling algorithm, namely FCMRPS, based on the traditional Shark-Search. Experiments for the focused crawlers show that the FCMRPS can outperform Breadth-First, Best-First and Shark-Search significantly in terms of precision and sum of information.

Essentially, our FCMRPS as well as most other focused crawling algorithms are greedy algorithms that can only find the most promising neighbors of the most relevant pages crawled so far. In other words, sometimes a focused crawler may be necessary to visit some slightly relevant or even irrelevant pages in order to arrive at the best ones. For example, if a good hub page is linked to irrelevant pages, and its anchor text and URL address are not explicitly relevant too, it is difficult for our algorithm to find it. Therefore, in the future we plan to take advantage of machine learning techniques based on historical information to solve this problem. If we repeatedly crawl a same portion of the Web about one topic, we have historical information available which is calculated from the previous crawling process. For example, we learn the URLs pointing to

good hub pages from the historical information as seed URLs. In addition, we try to study more techniques needed in focused crawling, e.g. the description of users' requires, page rank in focused crawling systems and the relevance predicting with domain knowledge.

Acknowledgements

This work was supported by the Specialized Research Fund for the Doctoral Program of Higher Education of China under grant 20070422107 and the Key Science-Technology Project of Shandong Province of China under grant 2007GG10001002.

References

[1] Brian D. Davison, Topical locality in the Web, in: Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR'00, ACM, New York, NY, USA, 2000, pp. 272–279.
[2] Soumen Chakrabarti, Mukul M. Joshi, Kunal Punera, David M. Pennock, The structure of broad topics on the Web, in: Proceedings of the 11th International Conference on World Wide Web, WWW'02, ACM, New York, NY, USA, 2002, pp. 251–262.
[3] Michael Hersovici, Michal Jacovi, Yoelle S. Maarek, Dan Pelleg, Menanchem Shtalhaim, Sigalit Ur, The shark-search algorithm: An application: Tailored Web site mapping, Comput. Netw. ISDN Syst. 30 (1–7) (1998) 317–326.
[4] P. DeBra, G. Houben, Y. Kornatzky, R. Post, Information retrieval in distributed hypertexts, in: Proceedings of the 4th RIAO Conference, 1994, pp. 481–491.
[5] Fangfang Luo, Guolong Chen, Wenzhong Guo, An improved "fish-search" algorithm for information retrieval, in: Proceedings of 2005 IEEE International Conference on Natural Language Processing and Knowledge Engineering, 30 Oct.–1 Nov. 2005, pp. 523–528.
[6] Zhumin Chen, Jun Ma, Jingsheng Lei, An improved shark-search algorithm based on link analysis, Journal of Computational Information Systems 3 (4) (2007) 1753–1758.
[7] Michael Chau, Hsinchun Chen, Comparison of three vertical search spiders, Computer 36 (5) (2003) 56–62.
[8] Junghoo Cho, Hector Garcia-Molina, Lawrence Page, Efficient crawling through url ordering, Comput. Netw. ISDN Syst. 30 (1–7) (1998) 161–172.
[9] Mohsen Jamali, Hassan Sayyadi, Babak Bagheri Hariri, Hassan Abolhassani, A method for focused crawling using combination of link structure and content similarity, in: Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence, WI'06, IEEE Computer Society, Washington, DC, USA, 2006, pp. 753–756.
[10] Filippo Menczer, Gautam Pant, Padmini Srinivasan, Miguel E. Ruiz, Evaluating topic-driven Web crawlers, in: Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR'01, ACM, New York, NY, USA, 2001, pp. 241–249.
[11] Filippo Menczer, Gautam Pant, Padmini Srinivasan, Topical Web crawlers: Evaluating adaptive algorithms, ACM Trans. Internet Technol. 4 (4) (2004) 378–419.
[12] P. Srinivasan, F. Menczer, G. Pant, A general evaluation framework for topical crawlers, Inf. Retr. 8 (3) (2005) 417–447.
[13] J. Johnson, K. Tsioutsiouliklis, C.L. Giles, Evolving strategies for focused Web crawling, in: Proceedings of the Twentieth International Conference on Machine Learning, ICML, Washington DC, 2003.
[14] Taher H. Haveliwala, Topic-sensitive pagerank, in: Proceedings of the 11th International Conference on World Wide Web, WWW'02, ACM, New York, NY, USA, 2002, pp. 517–526.
[15] TREC (Text REtrieval Conference) http://trec.nist.gov/pubs/trec12/papers/WEB.OVERVIEW.pdf.
[16] CWT (Chinese Web Test) http://www.cwirf.org/Evaluation/CWT.html.
[17] Zhumin Chen, Jun Ma, Xiaohui Han, Dongmei Zhang, An effective relevance predation algorithm based on hierarchical taxonomy for focused crawling, in: Proceedings of the 2008 Asia Information Retrieval Symposium, in: LNCS, vol. 4993, Springer, 2008, pp. 623–629.
[18] Soumen Chakrabarti, Kunal Punera, Mallela Subramanyam, Accelerated focused crawling through online relevance feedback, in: Proceedings of the 11th International Conference on World Wide Web, WWW'02, ACM, New York, NY, USA, 2002, pp. 148–159.
[19] Soumen Chakrabarti, Martin van den Berg, Byron Dom, Focused crawling: A new approach to topic-specific Web resource discovery, Comput. Netw. 31 (11–16) (1999) 1623–1640.
[20] Marc Ehrig, Alexander Maedche, Ontology-focused crawling of Web documents, in: Proceedings of the 2003 ACM Symposium on Applied Computing, SAC'03, ACM, New York, NY, USA, 2003, pp. 1174–1178.
[21] Michelangelo Diligenti, Frans Coetzee, Steve Lawrence, C. Lee Giles, Marco Gori, Focused crawling using context graphs, in: Proceedings of the 26th International Conference on Very Large Data Bases, VLDB'00, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2000, pp. 527–534.
[22] Weizheng Gao, Hyun Chul Lee, Yingbo Miao, Geographically focused collaborative crawling, in: Proceedings of the 15th International Conference on World Wide Web, WWW'06, ACM, New York, NY, USA, 2006, pp. 287–296.
[23] Open Directory Project http://dmoz.org/.
[24] Yahoo! directory service. http://search.yahoo.com/dir.
[25] Google directory service. http://www.google.com/dirhp?hl=en.
[26] F. Tanudjaja, L. Mui, Persona: A contextualized and personalized Web search, in: Proceedings of the 35th Annual Hawaii International Conference on System Sciences (HICSS'02), HICSS'02, IEEE Computer Society, Washington, DC, USA, 2002, pp. 1232–1240.
[27] Paul Alexandru Chirita, Wolfgang Nejdl, Raluca Paiu, Christian Kohlschütter, Using ODP metadata to personalize search, in: Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR'05, ACM, New York, NY, USA, 2005, pp. 178–185.
[28] B.Y. Ricardo, R.N. Berthier, Modern Information Retrieval, ACM Press, 1999.
[29] Charu C. Aggarwal, Fatima Al-Garawi, Philip S. Yu, Intelligent crawling on the World Wide Web with arbitrary predicates, in: Proceedings of the 10th International Conference on World Wide Web, WWW'01, ACM, New York, NY, USA, 2001, pp. 96–105.
[30] CWT200G(Chinese Web Test collection with 200 GB Web pages). http://www.cwirf.org/SharedRes/DataSet/CWT200g/CWT200g_number.
[31] B. Pinkerton, Finding what people want: Experiences with the Web crawler, in: Proceedings of the 2nd International World Wide Web Conference, USA, 1994.
[32] Marc Najork, Janet L. Wiener, Breadth-first crawling yields high-quality pages, in: Proceedings of the 10th International Conference on World Wide Web, WWW'01, ACM, New York, NY, USA, 2001, pp. 114–118.